

Trabalho 2

Introdução ao Processamento Digital de Imagem (MC920/MO443)
Professor: Hélio Pedrini

Vinicius Couto Espindola | RA: 188115

30 de Março de 2020

1 Introdução

Filtros são amplamente utilizados em processamento digital de imagens, estes visam passar, modificar ou rejeitar certos aspectos da imagem. São muito utilizados em aprimoramento de imagens dentre outras finalidades. O trabalho tem foco em filtros de imagens no domínio espacial, estes são compostos de operações que modificam os pixels de uma imagem a partir de funções aplicadas sobre o pixel e seus vizinhos. Para tal operação, utiliza-se um núcleo (matriz de constantes) o qual define, além do tamanho da vizinhança, as constantes aplicadas sobre os pixels da vizinhança. Neste trabalho, aplicaremos 8 filtros no domínio espacial utilizando a operação de convolução.

2 Execução

O programa requer as seguintes bibliotecas: *OpenCV* e *NumPy*. Sua execução pode tanto exibir a imagem de saída, quanto salvar o resultado em um arquivo. Todos os resultados produzidos e utilizados neste resultado encontram-se na pasta `./outputs`. As imagens de entrada utilizadas estão na pasta `./inputs`.

```
assignment_2.py [-h] <input> <kernel> [--sum <kernel>] [-o <output>]
```

input: *Path* da imagem de entrada a ser processada.

kernel: Deve ser um inteiro de 1 a 8, os quais correspondem aos filtros definidos no enunciado do trabalho.

Nota: O `< kernel >` corresponde ao índice "*x*" definidos no enunciado do trabalho e no decorrer deste relatório. Tais índices estão indicados como h_x .

sum: Recebe um segundo `< kernel >`. Quando *sum* for setado, ambos os filtros serão aplicados à imagem de entrada e o resultado destes serão somados de acordo com a função: $out = \sqrt{(h_x)^2 + (h_y)^2}$.

output-file: *Path* onde a imagem de saída deve ser salva.

Nota: Caso não seja definido um *output-file*, a imagem será exibida na tela. Se for definido, a imagem de saída será salva apenas.

3 Soluções

Para aplicar os filtros de forma eficiente, foi utilizada a função `cv2.filter2D(imagem, profundidade, kernel)` em conjunto com a função `cv2.flip(kernel, axis)`. A primeira calcula a correlação da imagem em relação ao *kernel* dado. No caso, a profundidade foi setada para `-1`, assim, está será a mesma da imagem de entrada. Para aplicarmos convolução a partir da correlação, a função `cv2.flip` foi utilizada para espelhar o *kernel*. O valor *axis* setado como `-1` indica que o núcleo será espelhado em relação à ambos os eixos. Nota-se, também, que a função `cv2.filter2D` faz o tratamento de borda utilizando a técnica de *padding*, que considera uma margem de pixels nulos quando calculando a convolução para os pixels da borda da imagem.

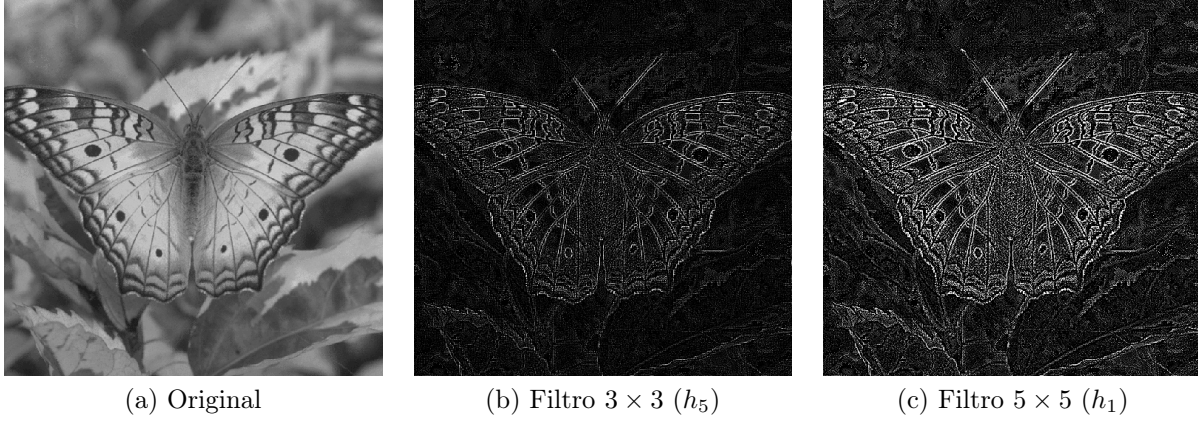


Figure 1: Resultados gerados pelos filtros passa altas.



Figure 2: Resultados gerados pelos filtros passa baixas.

3.1 Filtros Passa-Altas (h_1 e h_5)

O filtro passa altas tem como principal finalidade realçar certas características da imagem, como bordas e linhas, por exemplo. Quando aplicado, regiões de transição da imagem ficam em destaque, o que, em contrapartida, causa um aumento do ruído na imagem. O h_5 trata-se de um *kernel* mais simples, de tamanho 3×3 e constantes menores. Este enaltece os traços da imagem com mais suavidade. Já o *kernel* h_1 é o conhecido como *Laplacian of Gaussian*, possui tamanho 5×5 , com constantes maiores, fazendo com que este seja mais intenso e seja mais suscetível à ruído (como pode ser observado na **Figura 1**).

3.2 Filtros Passa-Baixas (h_2 e h_6)

O filtro h_6 trata-se de um *filtro da média* que atua como passa-baixas. Neste caso, temos um efeito de suavização da imagem, uma vez que regiões homogêneas não sofrem grandes alterações, já regiões de contraste, temos uma redução da diferença numérica entre os pixels. Nota-se, também, que o efeito é ameno, isso deve-se ao fato que o *kernel* pequeno é pequeno (3×3). Já o filtro h_2 consiste de uma função gaussiana com $\sigma = 1.0$. Aqui temos uma suavização ponderada dos pixels: quão mais próximo do centro do *kernel*, maior o peso do pixel. Isso possibilita uma suavização mais precisa, ou seja, que preserve as bordas da imagem de maneira mais eficiente. Pela **Figura 2**, nota-se que ambos são amenos, e a diferença entre eles quase imperceptível. Isso deve-se ao fato que o gaussiano possui um valor relativamente baixo de σ , e o da média é um filtro pequeno.

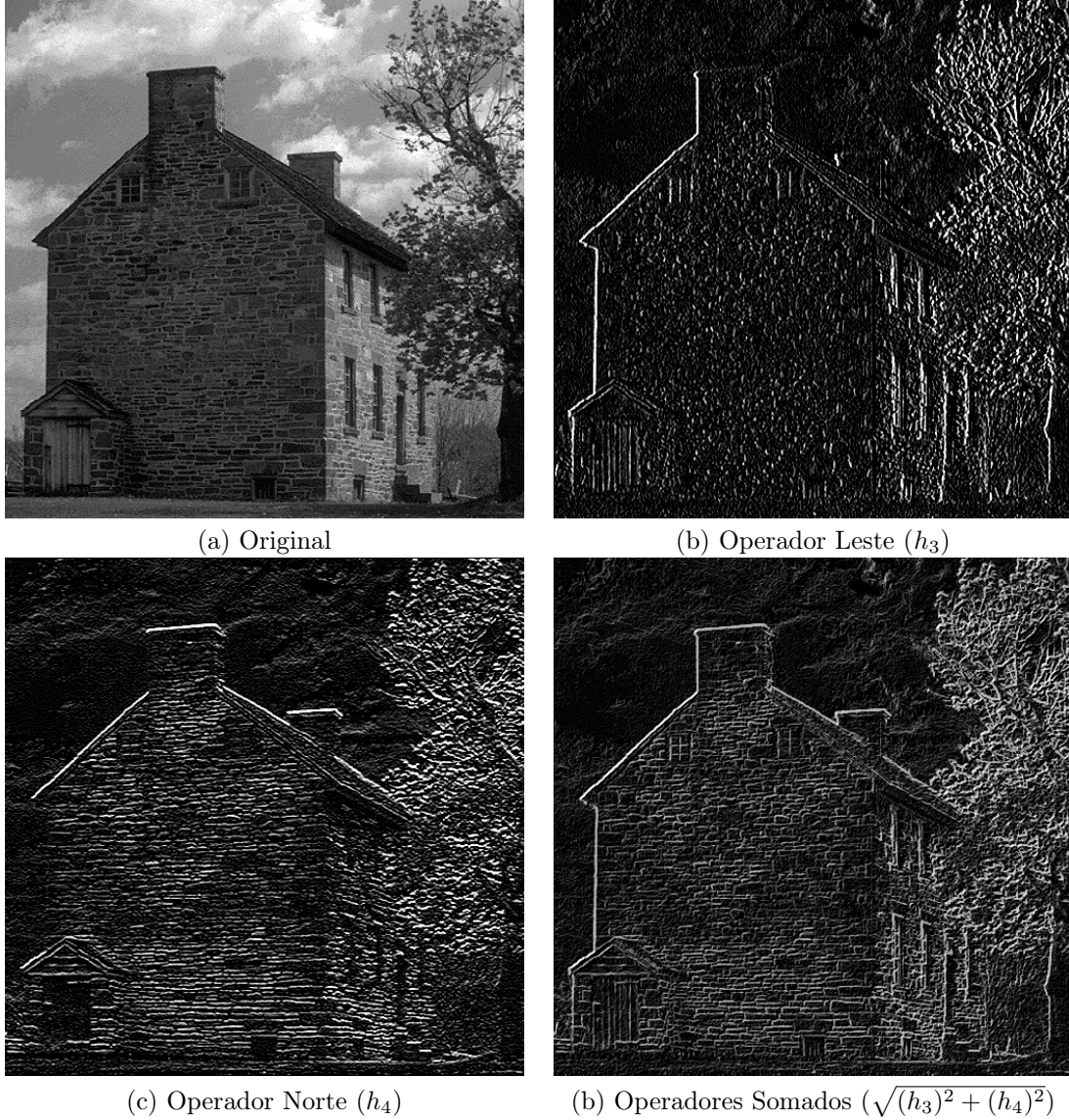


Figure 3: Resultados gerados pelos operadores de Sobel.

3.3 Operadores de Sobel (h_3 e h_4)

Os operadores de Sobel visam realçar contornos, no caso, temos um realce de arestas horizontais (h_4), mais especificamente, no sentido norte (pesos positivos em cima após a rotação do kernel), e arestas verticais (h_3) no sentido leste (pesos positivos à esquerda após a rotação do kernel). O efeito faz com que regiões homogêneas fiquem ocultadas, e regiões de transição, destacadas. No sentido leste (h_3), temos que o lado esquerdo de uma borda vertical tende a ser clareado e o direito escurecido. Já no sentido norte, nota-se o destaque das bordas horizontais com um clareamento da parte superior destas. Na **Figura 3**, temos o destaque das linhas verticais (b) e o das horizontais (c) nos batentes das janelas e nos tijolos da casa.

3.3.1 Junção dos Operadores

Ao somarmos os dois operadores (linhas horizontais e verticais), temos uma combinação que destaca os contornos da imagem. Em (d) temos contornos bem definidos a partir da combinação dos operadores.

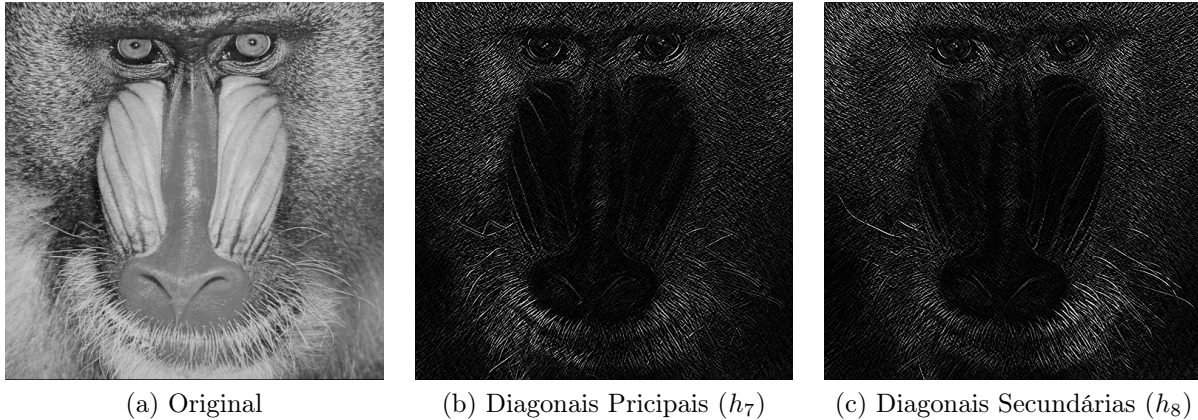


Figure 4: Resultados gerados pelos filtros passa altas.

3.4 Detecção de Arestas Diagonais (h_7 e h_8)

Nestes filtros, os *kenels* focam os pesos positivos nas diagonais: em h_7 estão na diagonal principal, e em h_8 , na secundária. Tal alocação permite um efeito de detecção de diagonais. Observa-se na **Figura 4** os pelos do babuíno são mais ressaltados de acordo com sua orientação: para o filtro h_7 temos que traços com cerca de 45° são destacados, já com o filtro h_8 , traços próximos à -45° são destacados.

3.5 Limitações

A função `cv2.filter2D` trata a maioria dos casos particulares que podem ocorrer durante a aplicação do filtro. No entanto, no caso de *overflow* do valor de um pixel, este será truncado para o maior valor aceitável, logo, espera-se que o filtro em questão tenha feito de forma a considerar tais situações. Exetua-se o caso da opção `-sum`, neste caso as variáveis são ajustadas para comportar valores maiores. Posteriormente, estes são normalizados para o intervalo $[0, 255]$. Vale ressaltar que filtros que não possuem uma célula central não são considerados.

4 Discussão e Conclusão

A possibilidade de aplicar filtros de forma a modificar certas características de uma imagem possui diversas aplicações. Para préprocessamento de dados, é fácil ver como estes podem ser utilizados em modelos de inteligência artificial para realce de detalhes relevantes. Outra aplicação clara de filtros é o aprimoramento ou modificação visual da imagem. Para redução de ruído, filtros do tipo passa-baixa (que aplicam o efeito de suavização) são essenciais. Em aparatos médicos também é comum ver filtros do tipo passa-altas que enaltecem traços como veias. A aplicação de filtros pode ter inúmeras finalidades.

References

- [1] R.C. Gonzalez. *Digital Image Processing*. Prentice Hall, 2007.